2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

Feature Selection based on Manifold-Learning with Dynamic Constraint Handling Differential Evolution

Zhihui Li[1], Zhigang Shang[1]
[1]School of Electrical Engineering
Zhengzhou University,
Zhengzhou, China
{lizhrain, zhigang_shang, liangjing}@zzu.edu.cn,

B. Y. Qu[2], J. J. Liang[1]
[2]School of Electric and Information Engineering
Zhongyuan University of Technology
Zhengzhou, China
qby1984@hotmail.com

***Abstract*—Feature Selection in high dimensional feature space is the main challenge in statistic learning field. In this paper, a novel feature selection method based on manifold learning is proposed. The distance metric weight vector are optimized to maximize the multi-class margin in the manifold embedded in low dimension space, as well as minimize its L1-norm. This multi objectives optimization problem is solved by a Differential Evolution (DE) with dynamic constraint - handling mechanism. And a criterion to determine the best feature subset based on the optimal weight vector is given. The test result for selecting the optimal feature subset of UCI breast tissue dataset indicates that this real coded feature selection method could find some feature subset which has good classification robustness.**

Keywords—Feature Selection; Manifold-Learning; Dynamic Constraint; Differential Evolution

I. INTRODUCTION

Feature selection means selecting a subset of d features from a set of D dimensional features space, generally d<D. There are three main objective of feature selection: improving the prediction performance of the features, providing faster or cost-effective features, and providing a better understanding of the underlying process that generated the data. With the feature dimension increasing more and more in many practice problem, the feature selection in high-dimensional space need more new theory and methods [1, 2].

Feature selection mainly based on optimization criterion. Most feature selection methods aim to find the feature subset which has good classification ability. But in a high-dimensional feature space, there are only finite training sample, that doesn't reflect any statistic rules. So many classifying features may perform well on the training data, but few may generalize well [3-5]. A good feature subset should keep better classification correctness and keep the intrinsic structure separable relationship underlying the original data.

For the classification question of few sample in high dimension, support vector machine (SVM) is the most popular method to find the maximum-margin hyper plane which has the largest separation margin between the two classes, and embody optimal stability. It should be noted that the computation of margin is based on the used distance metric. It is possible that you will get totally different results if you use different distance metric[6].

Manifold learning (ML) is a popular approach to non-linear dimensionality reduction in recent decade[7, 8]. The idea of ML is the dimensionality of many data sets is only artificially high, and the relationship between the points in high dimension space will be hold in an embedded low dimension space. So if there lies a manifold in original high dimensional data, we can observe and analysis the same data structure information in a low dimensional space. In ML, the reasonable distance metric is the key factor. Here let's take the ISOMAP algorithm for example, as one of the typical approaches to manifold learning; it uses the geodesic distances metric to seek a lower-dimensional embedding which maintains the geodesic distances all points. An idea is generated naturally that since the manifold embedding in low dimension keeps the structure relation of data, so the project of original data maybe reflect some rule in statistic sense. So we can find the maximum-margin in manifold space, which should have better classification ability than in high dimensional feature space.

Recently, LASSO(Least Absolute Shrinkage and Selection Operator) methods are used to feature selection[9]. The main idea of LASSO is to minimize the L1-norm of feature weight vector. It shrinks some feature weight and sets others to zero and hence tries to retain the good features of both subset selection and ridge regression. This property is good for most feature selection problem. So in this work, the L1-norm minimization will be used as a constrained condition.

When the optimization objective function is created, in order to discover the most informative and least redundant feature subset among the whole feature space, the 'space search' strategies to select features is important. Swarm intelligence algorithms have been used widely [10-12]. The genetic algorithm (GA) shows advantages for feature selection in many published literatures. Binary coded Particle Swarm optimization (PSO) and Ant colony optimization (ACO) also have been utilized in feature selection for their promising performance to solve the combinatorial optimization problems. But the binary coded mode, in which one feature will be used or deleted before evaluating the optimization objective function, is too rigid to lose the complementary information inter-features. In our previous work, we proposed a real coded feature selection method to optimize the weight vector of distance metric to get the maximum class margin in original feature space based on the global distance metric learning, without any constrained condition [13].

In this paper, we will try to make the maximum class margin in embedding low dimensional manifold space, under the constraint of minimizing the L1-norm of the weight vector of distance metric. In order to solve this multi-objective optimization problem, a differential evolution algorithm with dynamic constraint-handling mechanism (DCDE) is used[14]. This method will be tested finally by the UCI breast tissue dataset. The rest of this paper is organized as follows: Section II outlines the concept of distance metric learning and ISOMAP method, and the optimization objective function is introduced in Section III. A detailed description of DCDE is given in Section IV. The section V gives the test experiment result of feature selection using the UCI dataset. At last, discussion and conclusion are given in Section VI.

II. Distance metric learning and ISOMAP

A. Distance Metric Learning

In M-dimension feature space, the distance between points $x \in R^M$ and $y \in R^M$ is defined as:

$$d_A^2(x,y) = \|x-y\|_A^2 = (x-y)^T A(x-y) \tag{1}$$

The typical problem of distance metric learning is the learning of the weighted matrix $A \in R^{M \times M}$. For some classifier like K Nearest Neighbor (KNN), its performance depends heavily on the distance metric weight information.

For a given collection of pair of similar and dissimilar points in supervised learning, a good distance metric will keep all the data pairs in the same class close while separating those in the different classes. So in order to learn a distance metric A from training data in the input feature space, a criterion to evaluate classification performance of A should be given first. Any linear manifold learning algorithm, which learns a linear transformation, in fact is equivalent to distance metric learning.

B. ISOMAP

As a global structure preserved nonlinear manifold method, ISOMAP use the geodesic distances metric substitute for tradition Euclidean distance of two points in feature space, and the distance matrix of all pair point are used with multidimensional scaling (MDS) to compute the reduced-dimensional positions of all the points.

The complete ISOMAP algorithm has three steps:

1) Construct neighborhood graph

The first step is to determine which points are neighbors on the manifold M and define the neighborhood graph. According to the distances $d_{X(i,j)}$ between pairs of points i, j in the input space X , the neighborhood relations can be represented as a weighted graph G over the data points with edges of weight $d_{X(i,j)}$ between neighboring points.

2) Compute shortest paths

Secondly, ISOMAP estimates the geodesic distances $d_{M(i,j)}$ between all pairs of points on the manifold M by computing their shortest path distances $d_{G(i,j)}$ in the graph G. the Floyd–Warshall algorithm often is used to compute the pair-wise distances between all other points.

3) Apply MDS and estimate the manifold

The final step applies classical MDS to the matrix of graph distances $d_{G(i,j)}$, constructing an embedding of the data in a D-dimensional Euclidean space Y that best preserves the manifold's estimated intrinsic geometry.

The detail about ISOMAP could be seen in [8].

III. Optimization Objective Function

In our consistent research, we take the feature selection problem as the real coded optimization. We think feature selection is also a problem of distance metric learning. The contribution of each feature for classifying or clustering both could be represented with the distance metric weight vector. In the optimization process, features weights are adjusted adaptively for each training point to reflect the importance of features in determining its class label, and finally we can set a threshold of weights vector to determine which features should be selected. So we can use real coded optimization algorithm to get the optimal or sub-optimal feature weights.

The distance metric weighted matrix A could be represented by:

$$A = diag(W) = \begin{bmatrix} w_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & w_m \end{bmatrix} \tag{2}$$

Where $W = [w_1, \cdots, w_m]$ is weight vector to be optimized, in which each element represents the weight of corresponding feature.

We still use the class margin as the evaluation criterion of classification performance of distance metric matrix A. Given a distance function, the two nearest neighbors of any sample X_n, one is in the same class (called nearest hit or NH) and the other in the different class (called nearest miss or NM), could be found. So the class margin of X_n is defined as:

$$CM = d_{AM}^2(X_n, NM_{X_n}) / d_{AM}^2(X_n, NH_{X_n}) \tag{3}$$

Clearly, the distance metric A will be a good choice when the corresponding CM is larger. It means the large between- class distance and the small within-class distance. In this study, we don't estimate the class margin in the original input space, but in the embedding manifold space, so the character M in subscript represents manifold found by ISOMAP. If there is no embedding manifold in high dimensional input space, so the original feature space could be taken as manifold space.

In order to get the robust classification, the global multi-class margin is used:

$$M_{M\,mc} = \min(CM_{M\,\min}^{1}, \cdots, CM_{M\,\min}^{i}, \cdots, CM_{M\,\min}^{c}) \quad (4)$$

Where superscript i means the ith class in totally C classes, and $CM_{M\,\min}^{i}$ is the minimum class margin from all points in the same class i in manifold space.

It is expected to find the optimal features weight vector to maximize the multi-class margin meanwhile under the considering of using the minimum number of features. So the Optimization Objective Function can be given as:

$$\max_{W} \left\{ \min(CM_{M\,\min}^{1}, \cdots, CM_{M\,\min}^{i}, \cdots, CM_{M\,\min}^{c}) \right\}$$
$$subject \quad to: \quad \min_{W} L_{1-norm}(W) \quad (5)$$

For this multi-objective optimization problem, we use a DCDE method we proposed with dynamic constraint-handling mechanism. The detail of this algorithm is seen in next section.

IV. Differential Evolution with Dynamic Constraint-Handling

As the DE algorithm is essentially a problem-solving unconstrained optimization method, so in order to solve constrained optimization problems, Differential Evolution with Dynamic constraint-handling (DCDE) algorithm using rules-based constraint handling mechanism to restate the single-objective constrained optimization as a set of single-objective unconstrained problems, then the constrained optimization will be much more simple. The objective and constraints are assigned to the individuals adaptively as its fitness according to which has the maximal pressure needed to be optimized to guide the population to search for feasible region.

The algorithmic description of DCDE is as follow:

A. Dynamic and random DE (drDE)

a) Initialization

The initial population chosen randomly with a uniform distribution for all the individuals within the search space constrained by the prescribed minimum and maximum parameter bounds.

For example, the initial value of the jth parameter in ith individual is generated by

$$x_j = x_{\min} + rand(0,1) \cdot (x_{\max} - x_{\min}) \quad j = 1, ..., D \quad (6)$$

Where rand(0,1) is a random variable with uniform distribution and bounded by [0,1] .

b) Mutation

For each mutation operation, the modal is selected randomly from the strategy pool which include the strategy "DE/rand/2" "DE/best/2" and "DE/rand to best/2" as listed in Table I. drDE randomly utilize difference strategy, For each target vector x_i ($i = 1, 2, ..., NP$), a new mutation vector v_i is generated by adding the weighted difference to another vector, just as:

TABLE I. CHOOSEN DIFFERENTIAL STRATEGIES

Symbol	Differential Expression
DE/rand/2/bin	$v_i = x_{r1} + F(x_{r2} + x_{r3} - x_{r4} - x_{r5})$
DE/best/2/bin	$v_i = x_{best} + F(x_{r1} + x_{r2} - x_{r3} - x_{r4})$
DE/rand to best/bin	$v_i = x_i + F(x_{best} - x_i) + F(x_{r2} - x_{r3})$

c) Crossover

Crossover operation is to mix the mutated vector with the target vector to yield the so-called trial vector. The trial vector can be generated by：

$$u_{ij} = \begin{cases} v_{ij} & if(rand(0,1) \le CR) or (j = j_{rand}) \\ x_{ij} & if(rand(0,1) > CR) or (j \ne j_{rand}) \end{cases} \quad (7)$$

In drDE, for each crossover, CR is randomly selected from array [0.9,0.3] or [0.3,0.2,0.1] in different stage of the algorithm, j_{rand} is a randomly chosen index $\in [1, D]$ which can ensure at least one parameter be copied from v_i.

d) Selection

The greedy criterion is used to determine whether the trail vector u_i generated in generation G come into the population of next generation (G+1) or not. It is selected by using our followed mechanism.

B. Dynamic constraint-handling mechanism

Suppose that there are m constraints, then the number of objectives needed to be optimized is $m+1$, i.e. $F = [f(x), g_1(x), g_2(x), ..., g_m(x)]$,

Define the pressure

$$p_i = \frac{\sum_{j=1}^{NP} (g_i(\mathbf{x}_j) > 0)}{NP} \quad , i=1,2,...,m \quad (8)$$

$$fp = 1 - \bar{\mathrm{p}}, \mathbf{p} = [p_1, p_2, ..., p_m] \quad (9)$$

$$gp_i = \bar{p} \cdot (p_i / \sum_{i=1}^{m} p_i) \quad (10)$$

thus $$fp + \sum_{i=1}^{m} gp_i = 1$$

The first step is using roulette selection according to fp and gpi to assign the objective function or a single constraint as a target to each individual. Then according to the target to do the optimize operations.

While updating individual, the following comparison criteria is used：

a) If the target of optimization is kth constraint $g_k(\mathrm{x})$ *and the constraint hasn't been satisfied, the offspring wins if*

$$\begin{aligned} & g_k(\mathrm{u}_i) < g_k(\mathbf{x}_i) \\ & \text{or, } sumg(\mathrm{u}_i) < sumg(\mathbf{x}_i) \\ & \text{or, } sumg(\mathrm{u}_i) = sumg(\mathbf{x}_i) \;\&\; f(\mathrm{u}_i) < f(\mathbf{x}_i) \end{aligned} \quad (11)$$

Where:

$$sumg(\mathbf{x}_i) = \sum_{k=1}^{m} (ratio_k \cdot (g_k(\mathbf{x}_i) > T_k) \cdot g_k(\mathbf{x}_i)) \quad (12)$$

$$ratio_k = \frac{1/g_k \max}{\sum_{k=1}^{m}(1/g_k \max)} \quad k = 1,\ldots,m \quad (13)$$

$$T_k = 0.5 \cdot (1 - \frac{fitcount}{0.5 \cdot Max\_FEs}) \cdot g_k \max \quad (14)$$

(Max_FEs: Max fitness evaluations, stop criterion)

The effect of Ratio is to balance the impacts of different constraints. The goal of setting parameter T is to find an acceptable region (AR) with a relaxed constraint which include the feasible region (FR) with a strict constraint, such that the AR can be shrunken .

In this way, the constraints that are more difficult will have more individuals work for it, while the easier ones will have less or even no individual working for it. So the search will focus on finding feasible solutions firstly, and then concentrate on improving the objective function. There will be more individuals evolve along the fitness increasing direction if more constraints are satisfied.

b) If the target of optimization is kth constraint $g_k(\mathrm{x})$ *and the constraint has been satisfied, the offspring wins if*

$$sumg(\mathrm{u}_i) < sumg(\mathbf{x}_i) \quad (15)$$

$$\text{or, } sumg(\mathrm{u}_i) = sumg(\mathbf{x}_i) \;\&\; f(\mathrm{u}_i) < f(\mathbf{x}_i) \quad (16)$$

c) If the target of optimization is $f(\mathrm{x})$, the offspring wins *if*

$$sumg(\mathrm{u}_i) < sumg(\mathbf{x}_i) \quad (17)$$

$$\text{or } sumg(\mathrm{u}_i) = sumg(\mathbf{x}_i) \;\&\; f(\mathrm{u}_i) < f(\mathbf{x}_i) \quad (18)$$

V. FEATURE SELECTION EXPERIMENT

In our numerical feature selection experiment, in order to test the performance of the method we proposed objectively, we use the exhaustive search method in small dataset first. The UCI Breast Tissue Data Set is used. This Dataset provides 106 instances, which including 9 impedance measurements features of breast tissue and 1 class attribute [15]. The dataset can be used for predicting the classification of 4 classes by merging together the fibro-adenoma, mastopathy and glandular classes whose discrimination is not important.

To get the robust feature subset which has a good classify capability and better generalization capability, we use the K-fold cross validation method to partition the dataset into 10 mutually exclusive sub-datasets. And the classifiers are used with KNN. For each training and test dataset, we use all possible 511 feature subset to make classification separately in original input feature space and in manifold space. Finally we can take some feature subsets which have larger median value and smaller range of classifying correct rate as candidate feature subsets. So we can compare the feature selection result using our method with these candidate feature subsets acquired by exhaustive search method.

All the algorithms are implemented using Matlab 2012a and executed on the computer with Intel Pentium® 4 CPU and 2Gb of RAM memory. For the convenience of denotation and comparison, we use the feature index to substitute name of original impedance features, showed as TABLE II.

TABLE II. SUBSTITUTION OF ORIGINAL IMPEDANCE FEATURES

Name	I0	PA500	HFS	DA	Area	A/DA	Max IP	DR	P
Index	F1	F2	F3	F4	F5	F6	F7	F8	F9

TABLE III. THE CANDIDATE FEATURE SUBSET WITH GOOD CLASSIFY AND GENERALIZATION CAPABILITY IN ORIGINAL OR MANIFOLD SPACE

Selected subset	**Correct Rate (median, range)**	**Space**
F1-F4-F7-F8	0.9000, 0. 1958	Input
F1-F4-F6	0.9045, 0. 2167	Input
F1-F3-F6-F7-F8	0.9000, 0. 2222	Input
F1-F3-F4-F7-F8-F9	0.9045, 0. 2000	Input
F1-F2-F4-F7-F9	0.9000, 0. 1591	Input
F1-F2-F4-F6-F7-F9	0.9045, 0. 2000	Input
F6-F9	0.9045, 0.1818	Manifold
F1-F4-F6-F7-F9	0.9045, 0.1091	Manifold
F1-F4-F6-F9	0.9083, 0.1818	Manifold

The candidate feature subsets selected by exhaustive search method are listed in TABLE Ⅲ. The median values of classifying correct rate used these selected feature subsets are larger than the 90% percentile of 511 median correct rates corresponding to all possible feature combination, at the same time their range of classifying correct rate are smaller than the 10% percentile of 511 range of classifying correct rate.

We use the K-fold(K=5) cross validation method to get the robust result of the optimization of objective function (5) with DCDE. For each training and test dataset, The population size is set to 100 and the maximum number of generation is set to 200.

When the weight vector representing contribution of each feature to classifying is achieved, features should be selected according to the following criterion:

$$w_k' = \frac{w_k}{\sum_{k=1}^{M} w_k}$$
$$\text{if} \quad w_k' > \frac{1}{M}, \quad \text{then} \;\; \mathrm{k}^{\mathrm{th}} \text{ feature } \textit{hold} \tag{19}$$

Where w_k is the kth element of optimal solution vector. When its normalized value is greater than the average weight of all m features, then the feature corresponding to it is be selected.

TABLE IV gives the optimal feature weights and final result of feature selection according to the criterion (Eq.5), for 5 different data partition for cross-validation. It indicates the features F1, F4, F6, F7, F9 are linked together frequently, and their combination make the classifier has robust and good performance at different data environment. The feature subset [F1 F4 F6 F7 F9] has the best performance and it is selected by the method we proposed. In this optimization process, the complementary information is preserved and the redundant relationship is eliminated between features. So the classification result using selected feature subset is high robust. This test result indicates that this real coded feature selection method could find some feature subset which has good classification robustness.

To evaluate the performance of our feature selection method, we use it to select the good feature subset in high dimensional dataset furthermore. The UCI SPECTF Heart Data Set is used [15], in which there are 267 instance with 44 image features. We compare our method (named as Robust Feature Selection, RFS) with other classic feature selection methods, including the Single Optimum Combination (SOC), Sequential Forward Selection (SFS), and standard genetic algorithm (GA). Their optimization objective function is maximization of the classifying correct rate. The result is seen in TABLE V. The result also indicates the most advantage of our method is the selected feature subset has a good robustness and has few number.

TABLE IV. THE WEIGHT OF FEATURES AND THE SELECTED FEATURE SUBSET

F1	F2	F3	F4	F5	F6	F7	F8	F9	Selected subset
0.86	0.47	0.64	0.99	0.52	0.77	0.32	0.46	0.85	[F1,F4,F6,F9]
0.12	0.03	0.09	0.03	0.05	0.17	0.11	0.20	0.20	[F1,F4,F7,F8,F9]
0.94	0.54	0.05	0.60	0.09	0.41	0.47	0.26	0.54	[F1,F2,F4,F7,F9]
0.16	0.03	0.07	0.14	0.11	0.18	0.13	0.01	0.17	[F1,F4,F6,F7,F9]
0.85	0.38	0.18	0.55	0.20	0.83	0.67	0.38	0.68	[F1,F4,F6,F7,F9]

TABLE V. THE COMPARATION RESULT OF DIFFERENT FEATURE SELECTION METHODS

Method	**Correct Rate(%) Mean±Std**	**Selected Feature Number**
RFS	90.1±4.1	17
SOC	88.4±9.2	32
SFS	90.2±5.9	22
GA	93.7±6.1	25

VI. CONCLUSION

Although these test result shows the feature selection with the algorithm we proposed is feasible and effective in high dimension, it is should noted here we only use the linear weighted idea as the first try. We will extend this algorithm to the nonlinear case by using kernel method for applying it to solve more complex and higher dimension problem. The future work will include more experiment, to investigate its efficacy compared to other conventional approaches, especially when handling large training data with high dimensions. Feature selection also can be handled as Multi-objective Optimization directly, and recently the Multi-swarm PSO which also real coded are emerging [16], we will try to integrated our objective function with these Multi-objective Optimization method to compare with the DCDE.

ACKNOWLEDGEMENT

This paper is supported by National Natural Science Foundation of China (U1304602), Scientific and Technological Project of Henan Province (122300410264), National Natural Science Foundation of China (61305080), Scientific and Technological Project of Henan Province (132102210521), Postdoctoral Science Foundation of China (Grants 20100480859) and Specialized Research Fund for the Doctoral Program of Higher Education (20114101110005).

REFERENCES

[1] Huan Liu; Lei Yu, "Toward integrating feature selection algorithms for classification and clustering," Knowledge and Data Engineering, IEEE Transactions on , vol.17, no.4, pp.491-502, April 2005

[2] Wenjie You, Zijiang Yang, Guoli Ji, Feature selection for high-dimensional multi-category data using PLS-based local recursive feature elimination, Expert Systems with Applications, Volume 41, Issue 4, Part 1, pp.1463-1475, March 2014

[3] Lazar, C.; Taminau, J.; Meganck, S.; Steenhoff, D.; Coletta, A.; Molter, C.; de Schaetzen, V.; Duque, R.; Bersini, H.; Nowe, A., "A Survey on Filter Techniques for Feature Selection in Gene Expression Microarray Analysis," Computational Biology and Bioinformatics, IEEE/ACM Transactions on , vol.9, no.4, pp.1106,1119, July-Aug. 2012

[4] David Dernoncourt, Blaise Hanczar, Jean-Daniel Zucker, Analysis of feature selection stability on high dimension and small sample data, Computational Statistics & Data Analysis, Volume 71, pp.681-693 March 2014,

[5] Ahmade, F.K.; Norwawi, N.M.; Deris, S.; Othman, N.H., "A review of feature selection techniques via gene expression profiles," Information Technology, 2008. ITSim 2008. International Symposium on , vol.2, no., pp.1,7, 26-28 Aug. 2008

[6] Zaidi, N.; Squire, D., "SVMs and data dependent distance metric," Image and Vision Computing New Zealand (IVCNZ), 2010 25th International Conference of , vol., no., pp.1,7, 8-9 Nov. 2010

[7] Lunga, D.; Prasad, S.; Crawford, M.M.; Ersoy, O., "Manifold-Learning-Based Feature Extraction for Classification of Hyperspectral Data: A Review of Advances in Manifold Learning," Signal Processing Magazine, IEEE , vol.31, no.1, pp.55,66, Jan. 2014

[8] A Global Geometric Framework for Nonlinear Dimensionality Reduction Joshua B. Tenenbaum, Vin de Silva, and John C. Langford Science 22 December 2000: 290 (5500), 2319-2323.

[9] Libal, U., "Feature selection for pattern recognition by LASSO and thresholding methods - a comparison," Methods and Models in Automation and Robotics (MMAR), 2011 16th International Conference on , vol., no., pp.168,173, 22-25 Aug. 2011

[10] Liu, H.: Evolving Feature Selection. IEEE Intell. Syst. 20(3), 64–76 (2005)

[11] Huang, H., Xie, H., Gup, J.: Ant Colony Optimization-based Feature Selection Method for Surface Electromyography Signals Classification. Computers in Biology and Medicine 42(2), 30–38 (2012)

[12] Zhang, J., Zhan, Z.: Evolutionary Computation Meets Machine Learning: A Survey. Computational Intelligence Magazine 6(4), 68–75 (2011)

[13] Zhigang Shang, Zhihui Li, J.J.Liang, “Real Coded Feature Selection Integrated with Self-adaptive Differential Evolution Algorithm,” Communications in Computer and Information Science, 2012 Eighth International Conference on Intelligent Computing (ICIC 2012), pp. 481-488, 2012

[14] Zhihui Li, J. J. Liang, Xi He and Zhigang Shang,, " Differential Evolution with Dynamic Constraint-Handling Mechanism, " IEEE Congress on Evolutionary Computation 2010, pp. 1899-1906, July, 2010

[15] Bache, K. & Lichman, M. (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

[16] Bing Xue; Mengjie Zhang; Browne, W.N., "Particle Swarm Optimization for Feature Selection in Classification: A Multi-Objective Approach," Cybernetics, IEEE Transactions on , vol.43, no.6, pp.1656,1671, Dec. 2013